X-CAMPUS: a proactive agent for ubiquitous services

Hajer SASSI[1], José ROUILLARD[1]

Abstract. Our research is at the crossroads of two main complementary areas: agents for intelligent systems and human-computer interactions. The prospect of highly natural interfaces, in which interaction between user and computer is mediated by lifelike agents, is currently one of the major steps leading to the acceptance of relevant and practical agents system by people. In this paper we illustrate our approach which takes into account both the evolution of the user interface technology and the emergence of ubiquitous computing in order to propose a proactive adaptive assistance through an agent named X-CAMPUS (eXtensible Conversational Agent for Multichannel Proactive Ubiquitous Services). It aims to assist proactively user in her daily tasks thanks to its ability to perceive the state of the environment and to communicate effectively according to the user's current situation.

Keywords: Intelligent Interfaces, ubiquitous computing, human-computer interaction, proactive assistance, multimodal interfaces, multichannel interfaces.

1 Introduction

Numerical world searches to make technologies more efficient through the development of system able to assist user according to her current situation thanks to the emergence of new devices able to communicate information that described user's environment. Many types of smart applications have recently appeared in order to help user reactively. This kind of applications requires user's intervention in order to satisfy her/his needs or thanks to ubiquitous devices and their capability to provide information which describe user his/her surroundings she/he can be satisfied without any explicit request.

We have decided to develop an approach which can determine user's needs based on contextual information that our agent has collected during the communication with the user.

The name "X-CAMPUS" chosen for our agent summarizes the principal characteristic that we have decided to study in our approach. X-CAMPUS stands for eXtensible Conversational Agent for Multichannel Proactive Ubiquitous Services:

[1] H. Sassi, J. Rouillard
NOCE team. Laboratoire d'Informatique Fondamentale de Lille, Bat M3 - F-59655 Villeneuve d'Ascq cedex.

- eXtensible: our system is not limited to only one application domain, but can be extended to many different ones (home automation, e-learnig, e-commerce, entertainment, tourism, etc.) according to the sensors available in order to capture the interaction context.
- Conversational: our system is able to dialog with the user in different conversational modes (menu, question/answer), in order to bring more natural communication between users and agents. It collects and provides information across several instant messaging and VOIP services (Gtalk, MSN, IPPI...).
- Agent: in our system, each user is connected to a digital agent. The main engine of our system (called Athena) is able to manage multiple conversations at the same time, and to trigger relevant actions according to a particular context. Some rules are used to decide each (simple or complex) action to perform according to various events (time, profile and location of the user, preferences, etc.).
- Multichannel: in our hyper-connected world, we have decided that X-CAMPUS would be able to communicate with users across multiple channels. We have particularly worked on Internet and phone channels, thus if a user is not connected online (Internet instant messaging), our system tries other possibilities (vocal phone call, SMS, E-mail...) in order to reach the user, proactively.
- Proactive: unlike most of the classical agent systems, mainly conceived on a reactive-based logical (user's request gives agent's response). X-CAMPUS was programmed with the possibility that the digital agent could push information to the user without any explicit solicitation.
- Ubiquitous: X-CAMPUS manages a range of heterogeneous devices in order to collect information and consequently to be able to assist user anywhere, anytime, in a completely ubiquitous and pervasive way.
- Services: our agent perceives user's environment permanently. Once it detects a change in the state of the user's surroundings it proceeds to analyze the new situation immediately and to determine if it should communicate information and provide relevant services to the user.

This paper is organized as follows: section 2 describes a walk-through example. In section 3 we detail our approach and we illustrate our agent's behavior in section 4. Section 5 reports some experimental results. Finally, Section 6 presents our conclusions and future work.

2 Our approach

The main goal of the use of ubiquitous devices is to ensure access to the state of our surroundings which formed user's current context. This term appeared in 1992 with Want and al. when they introduced their system named Active Badge Location System which is considered to be one of the first context-aware applications. It aims to build a system for phone calls delivery according to the called person's localization.

After that many other definitions were succeeded and most of them are responses to the following questions: why? What? Where? When? and how? That will generate an enormous quantity of useful and useless information [2]. Chen and Kotz [3] con-

cluded that existent definitions of context are very general, vague and not clear enough to help its understanding in computing systems. In the following we will give, in chronological order of appearance, a non-exhaustive list of the most relevant definitions of context which take into account explicitly the human-computer interaction.

In 2000, Dey defines context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [5]. Salber [15] described context as any information required by the interaction between the user and the application which can be sensed by the application. Miraoui and al. define context as "Any information that trigger a service or change the quality of a service if its value changes" [13]. In our work, to use context effectively we have decided to define context as any information that let our system initiate a conversation or change the way of interaction in order to ensure the continuity with the user.

3 Architecture

Context aware systems can be implemented in different ways according to systems their requirements and their conditions such as the location of sensors (local or remote), the amount of possible users (one user or many), the available resources of the used devices (high-end-PCs or small mobile devices) or the facility of a further extension of the system [1]. Furthermore, the method of context-data acquisition is very important when designing context-aware systems because it predefines the architectural style of the system at least to some extent. Chen presents three different approaches on how to acquire contextual information: direct sensor access, middleware infrastructure and context server [4].

In order to exploit pervasive information around us, we need to follow a context-model that define and store context data. Strang and Linnhoff-Popien summarized the most relevant context modeling approaches, which are based on the data structures used for representing and exchanging contextual information in the respective system [17]. So he classifies context-models in six models which are key-value model, markup scheme models, graphical models, object oriented models, logic based models and ontology based models. For technical reasons we have decided to model our context by using object-oriented techniques which offers to use the full power of object orientation (e.g., encapsulation, reusability, inheritance).

Our architecture is a framework developed on .net technologies based on three main layers. It principal vocation is to develop a digital ubiquitous agent. Once, agent is developed, it will be deployed on a server containing a database of contextual information (based on an object oriented model for context description), supporting various constraints and rules developed in an ad-hoc way. Our system has the capability to manage many users simultaneously; it can communicate with client (applications) through three different protocols which are Microsoft Notification Protocol (MSNP) [10], Extensible Messaging and Presence Protocol (XMPP) [9] and Simple Object Access Protocol (SOAP) [11]. Moreover of these protocols, our approach uses

a scripting language based on the mechanism of the Artificial Intelligence Markup Language (AIML) [8] in order to facilitate the development of conversational agent. This language named Athena [12] is composed of several units, the most relevant ones are: the pattern, the template, the label and the scenario. The first unit "pattern" is reserved about what user should say, whereas the second unit "template" is about what agent should response. The third unit "label" is reserved to trigger (initiate) a proactive conversation and the last unit "scenario" is a set of pattern/template.

3.1 Context Manager Layer

To build systems able to act differently according to context awareness, intelligent environment should perceive and control sensors networks regularly through the "context manager layer". This layer should communicate with heterogeneous sources in order to collect information and register them in the database. This layer is based on context provider and context repository. It controls the behavior of sensors and saves new issues values (static, dynamic and temporary information) in context repository.

Static information represents data which remain unchanged during a long period, such as user's name, surname, age, etc., which are communicated by user during the first exchange with X-CAMPUS. However dynamic information changes frequently (examples: user's calendar events, location…). X-CAMPUS communicates with a SOAP client who manages two logical sensors: Google Calendar and Google Latitude. These two dynamic sources of information are perceived continuously in order to ensure a best proactive assistance to the user. The last types of information managed by X-CAMPUS are those strongly related to the time (date and/or hour). It's for example the moment chosen by the users from the forecast weather service notification.

This layer communicates directly with the second layer in order to publish information even before context repository registers information in database for later use. An example of sensors that we used to collect information is a Radio Frequency Identification (RFID) reader accompanied with RFID tags. When RFID reader detects an RFID tag, it firstly determines the user's name in order to salute her and secondly calculates the number of persons at home [16]. To gather user's information (current activity and preferences), we have chosen to ask some questions according to the user's context among an xml file which contains some questions grouped by theme [16]. User can also enter data through a software entity (e.g., Website, Google calendar, Face- book, etc.) and provide access to system which can use this software in order to more help user. This layer distinguishes three types of information: the static information, the temporary information and the dynamic information. Static information remains unchanged during the process of learning (e.g., name, age, etc.). Temporary information can be sometimes changed (e.g., preferences, taste, etc.). However dynamic information changes frequently (e.g., location, mood). All these types of information are stored in a database in order to be used later.

3.2 Anticipate Needs Layer

In our research, we are based on "context manager layer" in order to anticipate user's services. In this layer, we try to exploit stored data context manager by associating a set of adaptive operators. Actually, we distinguish three types of operators [16]: Conversion operator which adapts the original format in order to associate a meaning manageable by the system. For example: when temperature sensor sends the raw data "2", the conversion operator interprets this value as "it's cold" or "it's hot", according to the real situation of the user. Extract operator which extracts only relevant information. Example: extract just the minute from the current time. Coupling operator used when system should aggregate various and heterogeneous (logical and/or physical) data. Thus we propose a coupling operator which tries to collect many data in order to "understand" non-trivial situations. For example detecting the location of users in a living room requires gathering information from multiple sensors throughout the intelligent home. It should also, in many cases, combine the results of several techniques such as image processing, audio processing, floor-embedded pressure sensors, etc., in order to provide valid information.

3.3 User Interface Layer

In a ubiquitous environment, the behavior of services does not depend just on explicit user interaction but also on environment's perceptions. Combing these two sources of information, system can better respond to user's expectations. Our system has to provide an adaptive way of interaction according to the user's situations.

The "user interface layer" should be able to define the context and choose the best way to interact by selecting the appropriate modalities and channels [16].

4 Illustration and scenarios

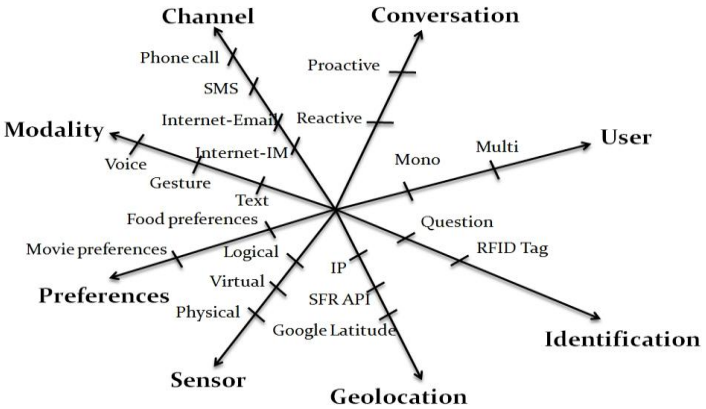

Fig. 1. X-CAMPUS's capabilities

As we can see in Figure 1, X-CAMPUS has many different skills which are represented by the different available axes. Thanks to the use of physical (e.g., RFID tags), virtual (e.g., Google calendar) and logical sensors (e.g., Google calendar combined with RFID tags), our system can interact in both reactive and proactive mode, according to user's needs. X-CAMPUS is also capable to manage some social aspects of the context. For example, events like "Eating at restaurant" registered in the user's agenda will lead to take into account the possible guests (friends, colleagues, etc.) involved in this particular event. Thus, some relatively complex and time consuming tasks can be done directly by a software agent that is in charge to find the better choice (example: the best restaurant) according to various criteria (alimentary preferences, distances and geolocation of each person, etc.). In order to ensure an adaptive interaction, we have decided to work on multi-channel and multi-modal interfaces and we have chosen to use two types of channels which are Internet and phone [14] and three types of modality which are text, gesture and voice.

We have proposed in [16] a scenario talking about favorite TV shows. With this scenario we have demonstrated the capability of our agent to collect relevant information concerning favorite TV categories of various users thanks to logical sensors (RSS with category, date, hour, duration…) which allows X-CAMPUS to identify user's favorite programs among many flows. It was also able to find the best way to communicate this information according to user's situation (at home, at office, alone, etc.)

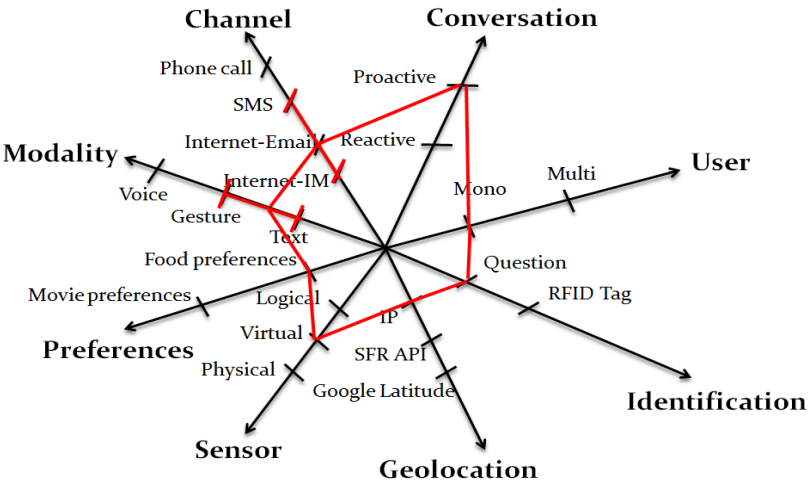

Fig. 2. X-CAMPUS's capabilities used in the favorite dishes

4.1 User's Favorite dishes

As we see in Figure 2, X-CAMPUS matches all axes by integrating one or more values for each capability. We will give more information about this figure in the section below.

Internet Channel.
This scenario is about Mr. Marc's favorite dish. System knows that Mr. Marc likes "potatoes, beef and pizza" and wants to be notified at 11 o'clock AM. The context used to satisfy user's favorite dish is: user's food preferences, user's favorite notification period, user's phone number, user's e-mail and restaurant menus. Every day, our system checks user's favorite notification period, menus proposed by restaurants and user's preferences dishes. If X-CAMPUS finds a minimum of one restaurant that contains a minimum of one of user's favorite dishes, it calculates the remaining time from the start period of notification and decides to send this information to the "User Interface Layer". Afterward, this latter layer sends a request to the "Context layer manager" in order to determine user's situation. For example, at the office and when user is connected, the system will provide this service using a classical text modality (MI) by sending information which contains the name of the restaurant, its menu, and on upper case user's favorite dishes through the Internet channel (see Figure 3). We can see that X-CAMPUS justifies its decision in order to let users understand easier its behaviors.

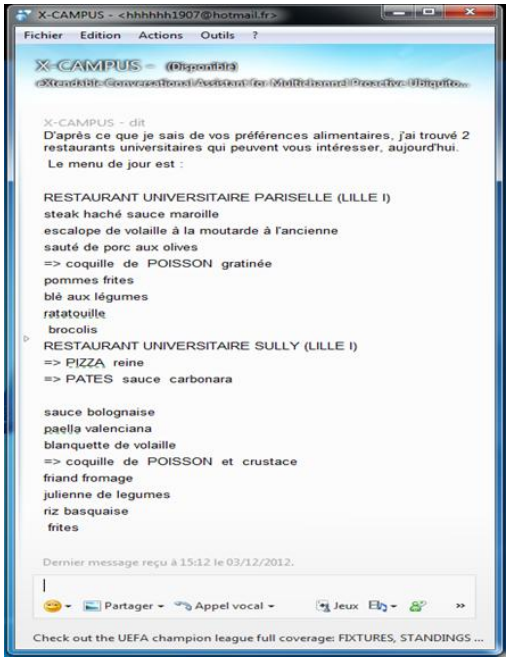

Fig. 3. X-CAMPUS notifies user about her favorite dishes

As we can see in figure below, X-CAMPUS has proposed two restaurants that their menu contains user's favorite dishes which are “Poisson”, “Pizza” and “Pates”. User can be disconnected from the Internet and in this case our system should find another way to communicate at the appropriate period/time. So thanks to a second channel that we use in our approach, service can be communicated in time through the phone-channel.

Phone Channel.

As we said in previous sections, we tried to provide proactive intelligent interfaces which can associate different types of modalities with different channels. However, when the user is disconnected from the internet network channel, and if the agent has important information to communicate to her, it should find a new way of communication to reach her wherever she is (home, office, outside, etc.). So, as second channel of communication that can be interesting in our work, we have chosen the phone channel, which allows our system to communicate with people when they are disconnected from the internet. This step is very important in our research; it ensures the continuity with the user by sending for example a Short Message Service message (SMS) as illustrated with Figure 4.

Fig. 4 Sending SMS through phone channel when user is disconnected

4.2 Complex Events

People frequently organize social events in order to meet, work, or discuss together. This kind of events are complex to managed (time and energy consuming) and to organized correctly. For example, going to the cinema, eating together, etc. can be some very difficult tasks to manage (for a human) as various and multiple criteria can be involved (time, weather, movie category, restaurant style, preferences of each person...).

Obviously, most of the time, the task in more and more complex as the number of invited is increasing. And sometimes, it becomes just impossible to find a convenient solution for everyone. In ordinary case, the organizer should call all participants or invite them through an electronic application such as Google Calendar or Doodle. As a consequence organizer should control by himself guests' responses and find the best solution which satisfies all participants.

In order to help users in these situations, X-CAMPUS has the capability to manage complex situations (multi-user, multi-modality, multi-channel...), according to user's situations, we have decided to integrate a new virtual sensor named "Google Calendar Sensor". Its vocation is to read user's Google Calendar and to send new events to our system. Once the system receives new events, it updates guests' profiles.

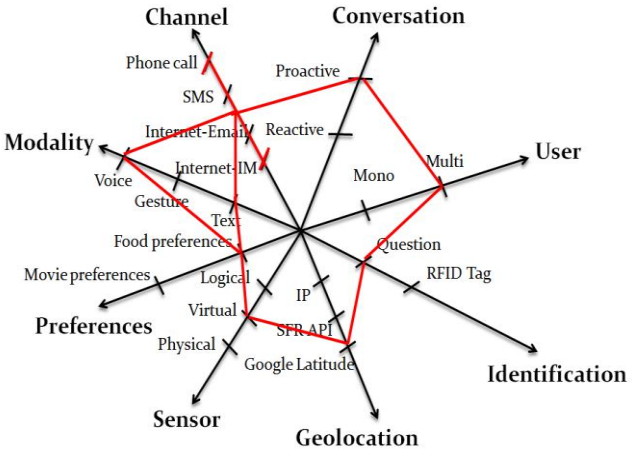

Fig. 5. X-CAMPUS's capabilities used in favorite social dish service

As we can see in Figure 5, X-CAMPUS will initiate a proactive conversation by considering as contextual information: user's mode (multi-user), user's location, user's food preferences, user's state (connected or disconnected), and also the possibility to change the used modality (text, voice) and channel (phone or Internet).

For example, a user (say Marc) decides to use Google Calendar to invite his two colleagues and to organize a meeting in a restaurant (criteria when and where). In this kind of situation, X-CAMPUS is switching from a non-social mode to a social mode. The difference between them is located at the "User Interface Layer". Once system decides to communicate service it checks if Marc has a social event named "Eating at university restaurant". According to system's check, it decides if it should behave in social or non-social mode. In the first case, we find ourselves with "user's favorite dish" whereas in the second case, system should check guests' favorite foods (except current user) thanks to the possibility to access to user's profile from her address mail (the address used to send invitation). Thereafter, it can determine guests' available restaurants more precisely the name of each restaurant that its menu contains at least one of guest's favorite dishes. Now, system knows each guest's available restaurants,

so it starts its analysis phase in order to find common restaurants. We worked on two different solutions to propose a restaurant to the organizer: a Boolean method and a more complex one (sorted list). With the first method, if a restaurant does not propose at least one favorite dish for a user, it is not selected. As we can see in Table 1, our system can find no common restaurants, one restaurant, or many restaurants which satisfy guests' needs.

	First day				Second day			
User / Restaurant	User #1	User #2	User #3	User #4	User #1	User #2	User #3	User #4
Restaurant #1	Pasta	Beef	Pizza	chips	Chiken			Fish
Restaurant #2	Pizza	Potatoes	Beef, pizza	Hamburger	Potatoes	Beef	Pizza	Pasta
Restaurant #3	Pizza, potatoes	Beef	Pizza	Pasta		Pizza	Fish	
	Three common restaurants				One common restaurant			

Table 1. X-CAMPUS's social behavior for two different days

The ideal situation is the case where only one restaurant satisfies all the users' preferences food (see Table 1). In this case it communicates by using a classical text modality through Internet channel or phone channel according to user's state. The message sent by the system contains the name of the chosen restaurant, the menu, the names of the guests and the user's favorite dishes (in uppercase). In a quite simple situation, X-CAMPUS is able to check, each day, for each user, the best suitable (university) restaurant to propose to the users according to various personal criteria stored in the user profile, such as alimentary preferences (Pizza, Beef, Pasta...), status (student, teacher...), etc. X-CAMPUS users are daily notified about the best match between their preferences and the different menus proposed by all the restaurants available in their geographical area. This notification is launched at a moment previously chosen. It can be done across an instant messaging tool (MSN, GTalk...), or by E-mail, or by SMS, according to the context of use. In a more complex manner, X-CAMPUS is also able to manage situations where people have planned to eat together: it calculates the more relevant restaurant(s) based upon the preferences of all the users involved in a particular meeting. This second resolution method tries to sort the restaurants, according the users' preferences, and is able to select the best solution, even degraded. Figure 6 and 7 present the logical optimization routine used to calculate the best list of restaurant to choose for a single user or for a group of users. The first example (Figure 6) is relatively simple, because the restaurant A is a solution for three users among five, and is naturally proposed to the organizer, as the best restaurant to choose, in this context.

		USER				
		1	2	3	4	5
RESTAURANT	A	1	1	2	2	3
	B	3	1	0	0	2
	C	1	0	3	1	3
	MAX	3	1	3	2	3
	Position max	2	1	3	1	1
	Choosen restaurant (by user)	B	A	C	A	A
Number of :	A	3				
Number of :	B	1				
Number of :	C	1				
Choosen restaurant for the group:	Position:	1				
	Value:	A				

Fig. 6. X-CAMPUS's decision (first situation)

In the second example (Figure 7), we can see that 4 preferred items of the user 1 (say for instance "chicken", "beef", "salad" and "pizza") are proposed by restaurant A.

In a "mono user" mode, the best suitable restaurant for user 1, 2, 3, 4 and 5 are respectively the restaurants A, C, B, C and B. In a "multi user" mode, our system has to propose the restaurant that maximizes the user constraints (favorite dishes, here). Thus, the restaurant B is proposed to the organizer because two users among five will have four favorite dishes in this restaurant.

		USER				
		1	2	3	4	5
RESTAURANT	A	4	1	2	0	3
	B	3	2	4	2	4
	C	1	3	3	3	3
	MAX	4	3	4	3	4
	Position max	1	3	2	3	2
	Choosen restaurant (by user)	A	C	B	C	B
Number of :	A	1				
Number of :	B	2				
Number of :	C	2				
Choosen restaurant for the group:	Position:	2				
	Value:	B				

Fig. 7. X-CAMPUS's decision in case two (second situation)

If a unique solution is available, then X-CAMPUS notifies directly all the users by indicating the chosen place. But if there is not only one solution, our system is able to trigger a communication with the organizer in order to choose across a conversation, the best contextual criteria to deal with this complex situation. Technically, when X-CAMPUS finds multiple solutions to a given situation (restaurants, movies, etc.) a vocal conversation is initiated with the organizer in order to take a decision. We are using Ippi Messenger [6] and Tropo [7] for this purpose. Ippi Messenger is a Voice Over IP (VOIP) tool, compatible with the Session Initiation Protocol (SIP). Tropo is a powerful yet simple API that adds Voice and SMS support to the programming languages that programmers already know (JavaScipt, PHP, Ruby, Python, Groovy...). In our example, when our system finds more than one restaurants, it decides to place a vocal call to the organizer. During this conversation, some other criteria are proposed to enlarge the contextual information set (geolocation of the participants, for instance).

5 Conclusion and Outlook

In this paper, we have presented our agent named X-CAMPUS which searches to increase the productivity and the welfare of the user situated in intelligent environments. We have illustrated our approach among a proactive multichannel and multimodal service in both simple (mono user) and complex way (social event). In the simple way X-CAMPUS searches to satisfy user according to her favorite dishes by proposing only menus of restaurants which contain user's favorite dishes. After that it analyses user's state in order to ensure the best way of interaction thanks to the notion of multimodality and of multichannel. Whereas in a complex way X-CAMPUS searches to satisfy more than one user simultaneously among two different solutions: a Boolean method and a more complex one (sorted list). Currently, we envisage an evaluation with users by proposing a set of proactive services in order to study users' behavior and our agent capabilities to adapt its behaviors according to user's context.

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007.
2. Brézillon, P., Borges, M.R., Pino, J.A., Pomerol, J.Ch.: Context-Awareness in Group Work: Three Case Studies. Proc. of 2004 IFIP Int. Conf. on Decision Support Systems (DSS 2004), Prato, Italie, Juillet, 2004.
3. Chen, G., Koltz, D.: "A survey of context-aware mobile computing research," Department of Computer Science, Dartmouth College, Technical report 2000.
4. Chen, H. "An Intelligent Broker Architecture for Pervasive Context-Aware systems", PhD Thesis, University of Maryland, Baltimore County, 2004.
5. Dey, A. K.: "Understanding and using context," Personal and ubiquitous computing, Vol. 5, pp. 4-7, 2001.
6. http://www.ippi.fr/index.php?page=home&lang=44&lang=44
7. https://www.tropo.com
8. http://www.alicebot.org/aiml.html
9. http://xmpp.org/xmpp-protocols/xmpp-extensions/
10. http://msnpiki.msnfanatic.com/index.php/MSN_Protocol_Version_18
11. http://www.w3schools.com/soap/soap_intro.asp
12. http://www.xbrainsoft.com/
13. Miraoui, M., et C. Tadj. 2007b. « A service oriented definition of context for pervasive computing ». In The 16th International Conference on Computing (November, 2007). Mexico city, Mexico: IEEE Computer Siciety.
14. Rouillard, J., Tarby, J-C., Le Pallec, X., Marvie, R. Facilitating the design of multi-channel interfaces for ambient computing, The Third International Conferences on Advances in Computer-Human Interactions, ACHI 2010, St. Maarten, Netherlands Antilles, pp. 95-100.
15. Salber, D.: "Context-awareness and multimodality", Colloque sur la multimodalité, Mai 2000, IMAG, Grenoble.
16. Sassi, H., Rouillard, J. Proactive Assistance within Ambient Environment, Intelligent Systems and Applications, pp. 60-65, INTELLI 2012, Chamonix.
17. Strang, T., Linnhoff-Popien, C. "A Context Modeling survey", In the first International Workshop on Advanced context modeling, Reasoning and management, UbiComp 2004.